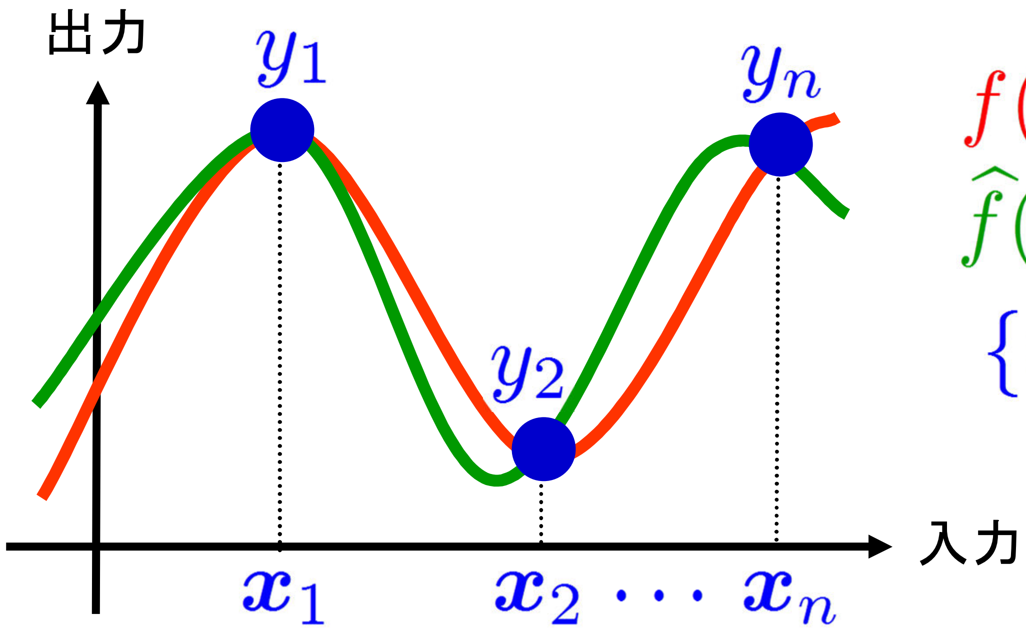
スパース回帰
（５章）

杉山将・本多淳也
sugi@k.u-tokyo.ac.jp, jhonda@k.u-tokyo.ac.jp
http://www.ms.k.u-tokyo.ac.jp

回帰 ＝ 関数近似



$f(\boldsymbol{x})$:学習したい真の関数

$\widehat{f}(\boldsymbol{x})$:学習結果の関数

$\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$:訓練標本

$$y_i = f(\boldsymbol{x}_i) \; (+\text{noise})$$

訓練標本から真の関数にできるだけ近い関数を求める

パラメータに関する線形モデル

■線形モデル: $f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sum_{j=1}^{b} \theta_j \phi_j(\boldsymbol{x})$

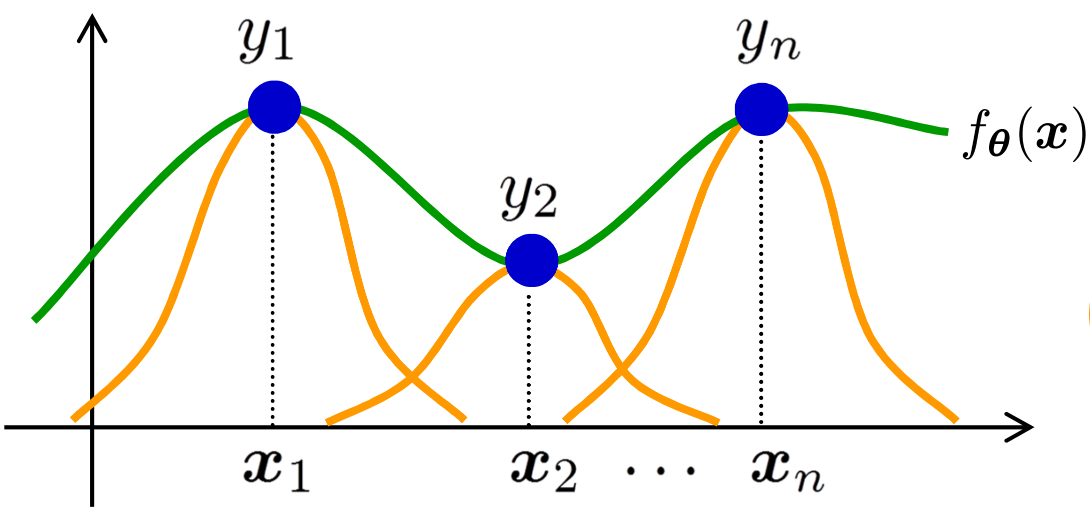
$\{\phi_j(\boldsymbol{x})\}_{j=1}^{b}$:基底関数

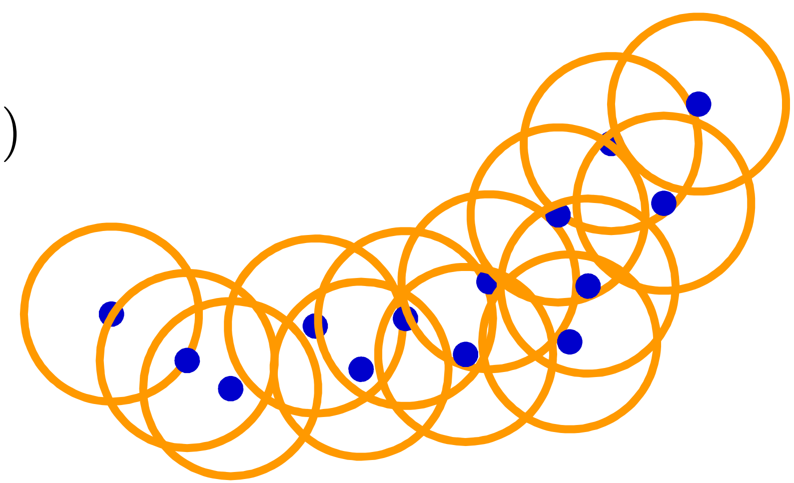
■カーネルモデル:

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sum_{j=1}^{n} \theta_j K(\boldsymbol{x}, \boldsymbol{x}_j)$$

ガウスカーネル

$$K(\boldsymbol{x}, \boldsymbol{c}) = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{c}\|^2}{2h^2}\right)$$

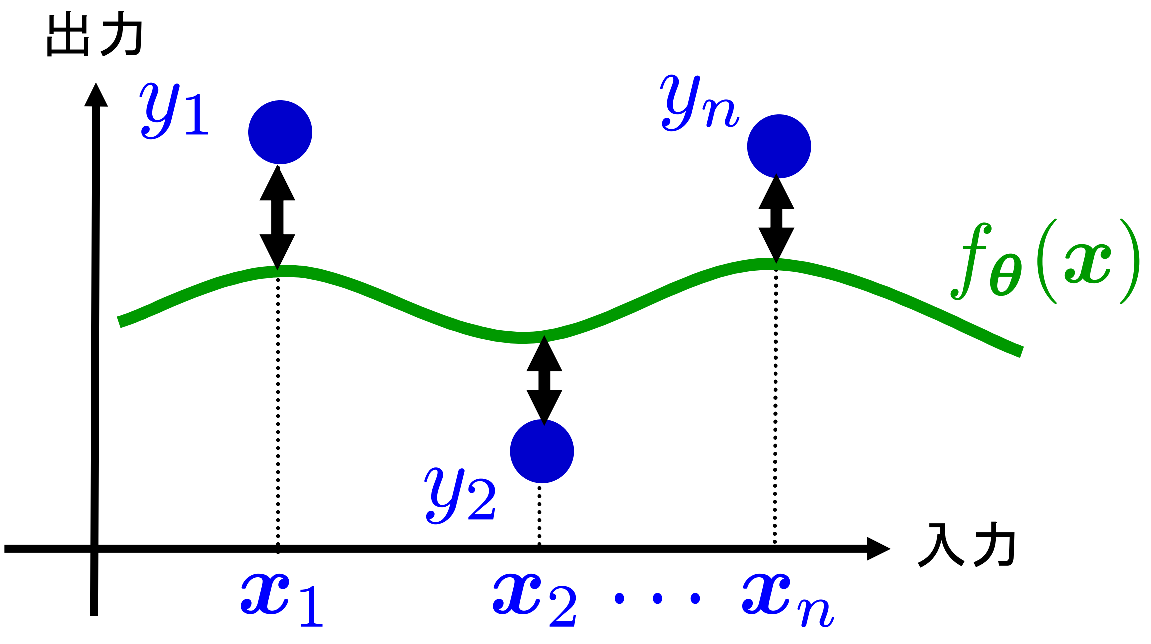




最小二乗回帰

- 訓練出力との二乗誤差を最小にする:

$$\min_{\boldsymbol{\theta}} \sum_{i=1}^{n} \left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) - y_i \right)^2$$



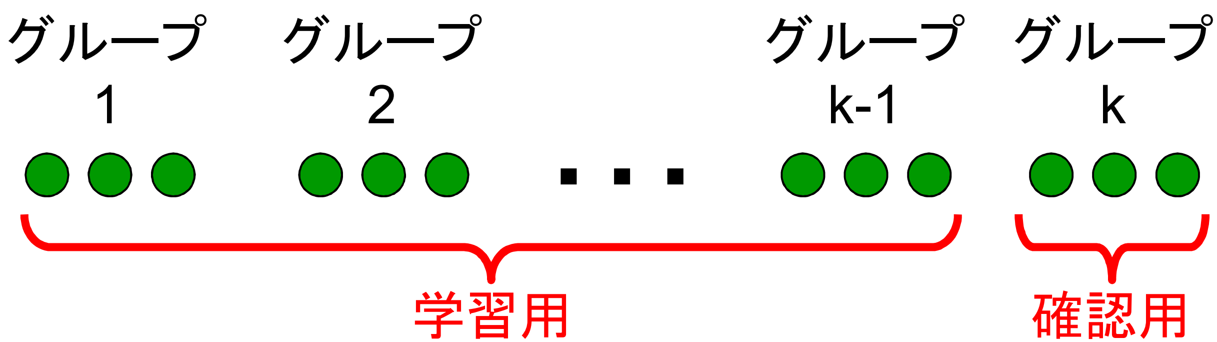
正則化

$$\min_{\boldsymbol{\theta}} \left[\underbrace{\frac{1}{2}\sum_{i=1}^{n}\Big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) - y_i\Big)^2}_{\text{訓練出力に対する適合の良さ}} + \frac{\lambda}{2}\underbrace{\|\boldsymbol{\theta}\|^2}_{\text{パラメータの値が大きくなり過ぎることに対する罰則（正則化）}} \right]$$

- 訓練出力に対する適合のよさとパラメータの値の大きさをバランスよく小さくし，過適合を避ける

交差確認によるモデル選択

- 訓練標本 $\mathcal{Z} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$ をk分割する: $\{\mathcal{Z}_i\}_{i=1}^{k}$
- $\mathcal{Z}_i$ 以外を使って関数を学習する
- 残った $\mathcal{Z}_i$ を使ってテスト誤差を確認する
- これを全ての組み合わせに対して繰り返し，平均を出力する



$\widehat{f}(\boldsymbol{x})$

$\left(\widehat{f}(\boldsymbol{x}') - y'\right)^2$

モデルのスパース性

- パラメータ数が多いと推定値の計算が大変
  - 例えばカーネルモデルでは,

    パラメータ数 ＝ 訓練標本数

    なので訓練標本数が多いとき計算が大変

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sum_{j=1}^{n} \theta_j K(\boldsymbol{x}, \boldsymbol{x}_j)$$

- 多くのパラメータ値がゼロになれば,
  出力値の計算が簡単になり, 解釈性も向上

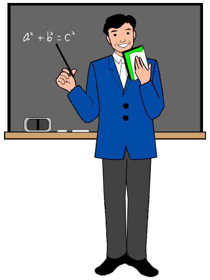
モデルのスパース性

ナイーブな方法1：

- あらかじめいくつかのパラメータを使用しないことにする
- パラメータが d 個ある時，選び方は 2^d 通り
- d が大きい時はとても網羅しきれない

ナイーブな方法2：

- ℓ_2-正則化回帰で得られたパラメータのうち，絶対値の小さいものをゼロに丸めてしまう
- 丸め誤差が発生する



講義の流れ

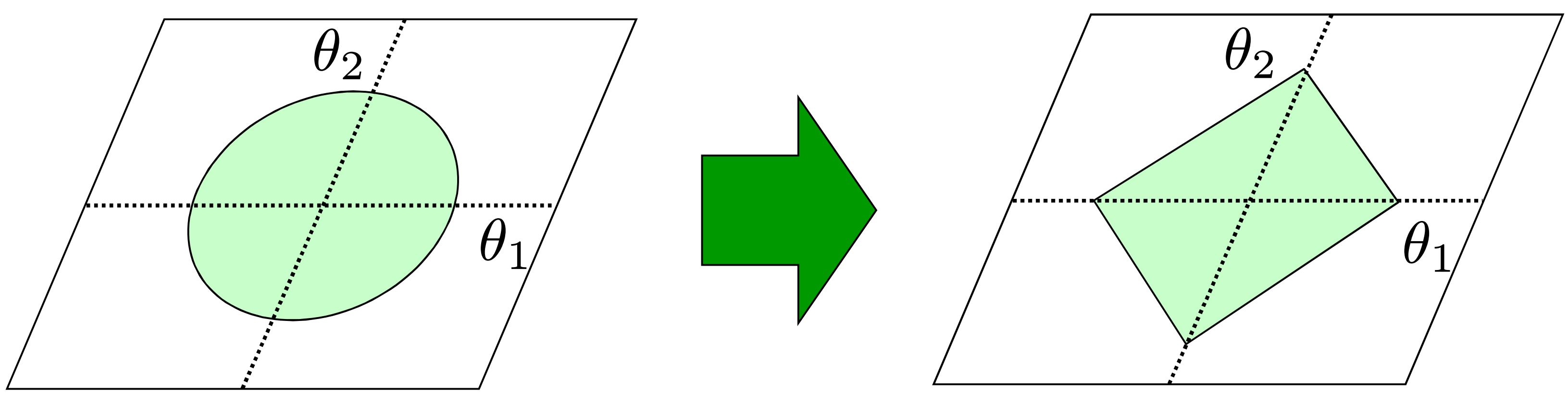
1. ℓ_1-制約付き最小二乗回帰
2. 解の計算法
3. 様々な拡張

ℓ_1-制約付き最小二乗回帰

■ モデルをℓ_1-超球に限定する

$$\min_{\boldsymbol{\theta}} \frac{1}{2}\sum_{i=1}^{n}\Big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) - y_i\Big)^2 \quad \text{subject to} \quad \|\boldsymbol{\theta}\|_1 \leq R$$

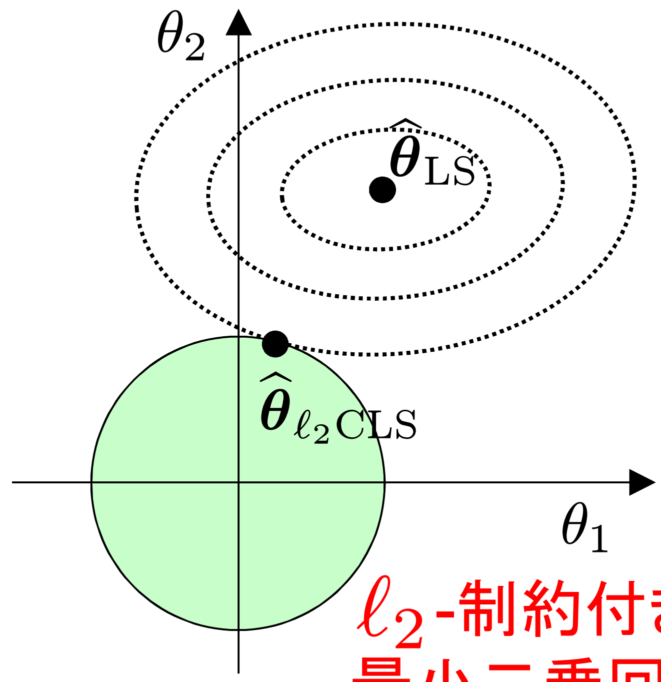
$$R \geq 0 \qquad \|\boldsymbol{\theta}\|_1 = \sum_{j=1}^{b} |\theta_j|$$



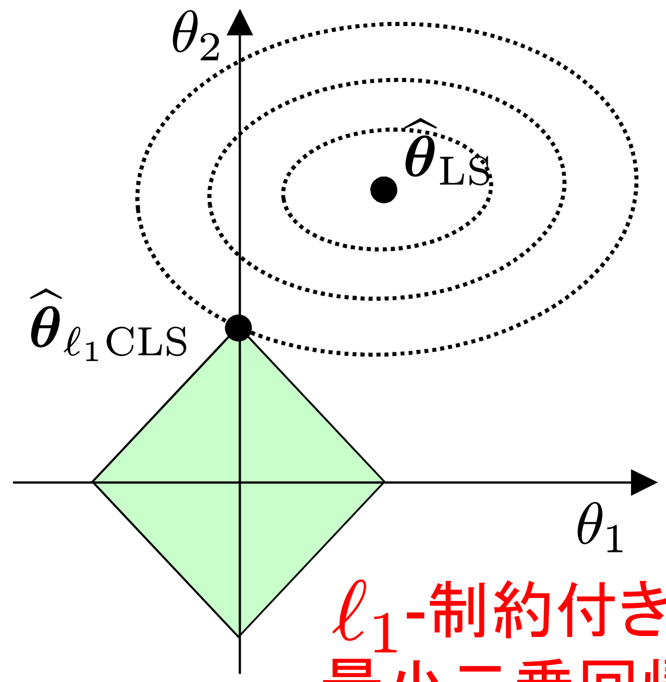
ℓ_2-制約付き最小二乗回帰　　ℓ_1-制約付き最小二乗回帰

なぜスパースな解が得られるか？

- 解が軸の上に乗ることが多い



ℓ_2-制約付き
最小二乗回帰

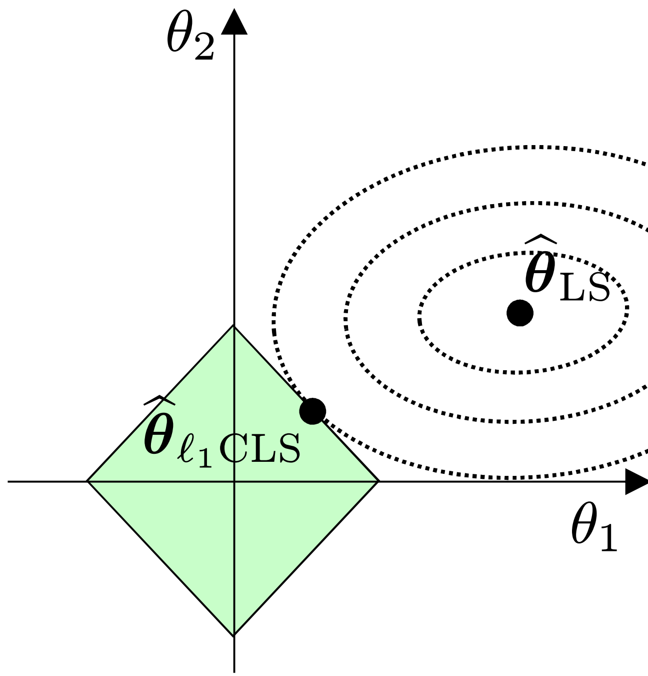


ℓ_1-制約付き
最小二乗回帰

- スパース回帰あるいは LASSO (Least Absolute Shrinkage and Selection Operator) と呼んだりする

なぜスパースな解が得られるか？

- 両方の変数が本質的に必要な場合は軸の上に乗らない



例

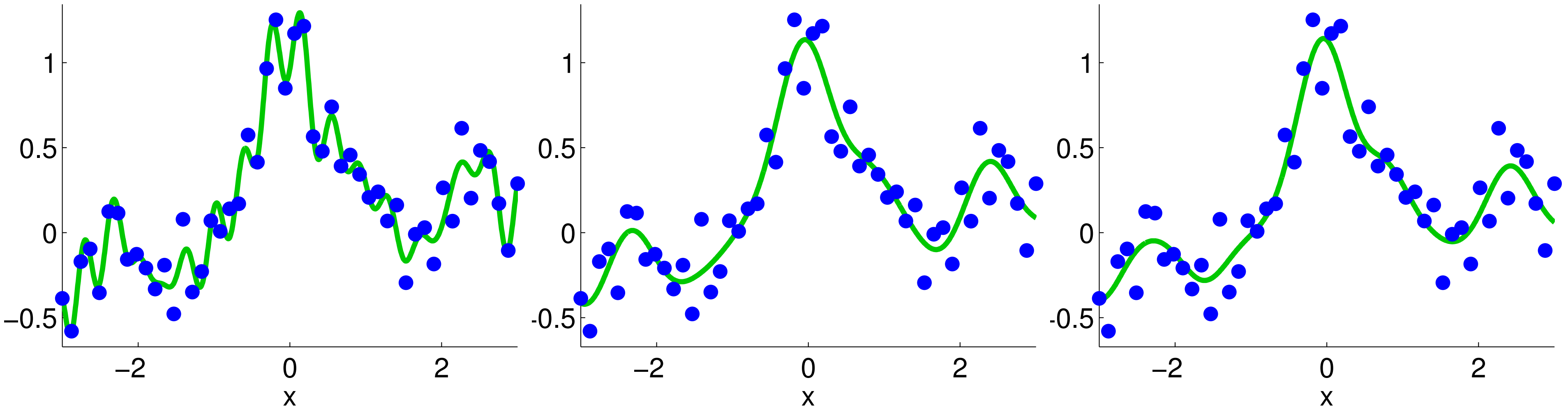
- ガウスカーネルモデル:

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sum_{j=1}^{n} \theta_j K(\boldsymbol{x}, \boldsymbol{x}_j)$$

$$K(\boldsymbol{x}, \boldsymbol{c}) = \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{c}\|^2}{2h^2}\right)$$

解の計算法は後ほど

通常の最小二乗回帰　　ℓ_2-制約付き　　ℓ_1-制約付き



- ℓ_1-制約の結果はℓ_2-制約と見た目は大体同じ
- しかし，ℓ_1-制約は50個のパラメータのうち38個が0

特徴選択

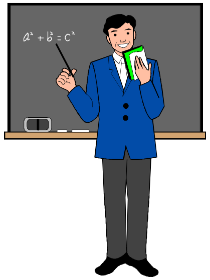
- 入力に関する線形モデル

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \boldsymbol{\theta}^{\top}\boldsymbol{x} \qquad \boldsymbol{x} = (x^{(1)}, \ldots, x^{(d)})^{\top}$$

に対してスパース学習を行なうと，いくつかの入力変数が使われなくなる．

予測に必要な特徴が自動的に選択される

- 例：重要な遺伝子の自動選択
- 通常は 2^d の組み合わせを調べる必要があるが，スパース学習では λ だけを決めればよい．



講義の流れ

1. ℓ_1-制約付き最小二乗回帰
2. 解の計算法
3. 様々な拡張

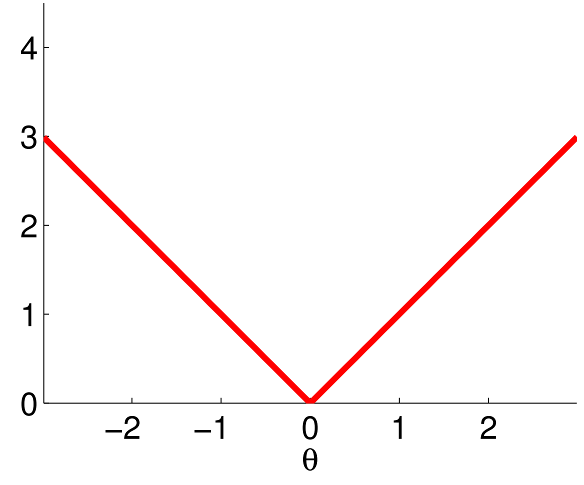
解の求め方

- 等価な表現:

$$\widehat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\mathrm{argmin}} \left[\frac{1}{2} \sum_{i=1}^{n} \left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) - y_i \right)^2 + \lambda \sum_{j=1}^{b} |\theta_j| \right]$$

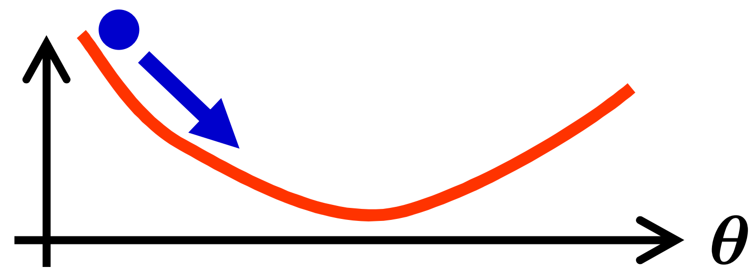
$\lambda\ (\geq 0)$: R から決まる定数

- 実際には R でなく λ を直接指定すればよい.
- しかし, 絶対値は原点で微分できないため, 上記の最適化問題は簡単には解けない

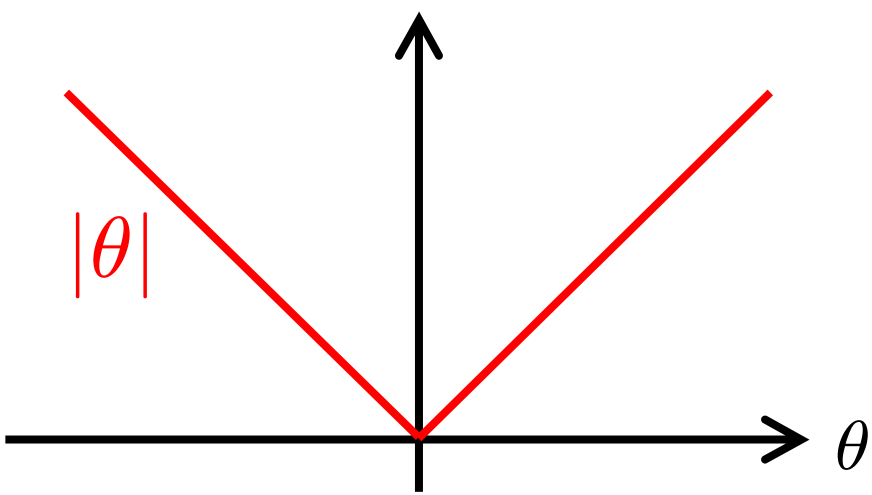


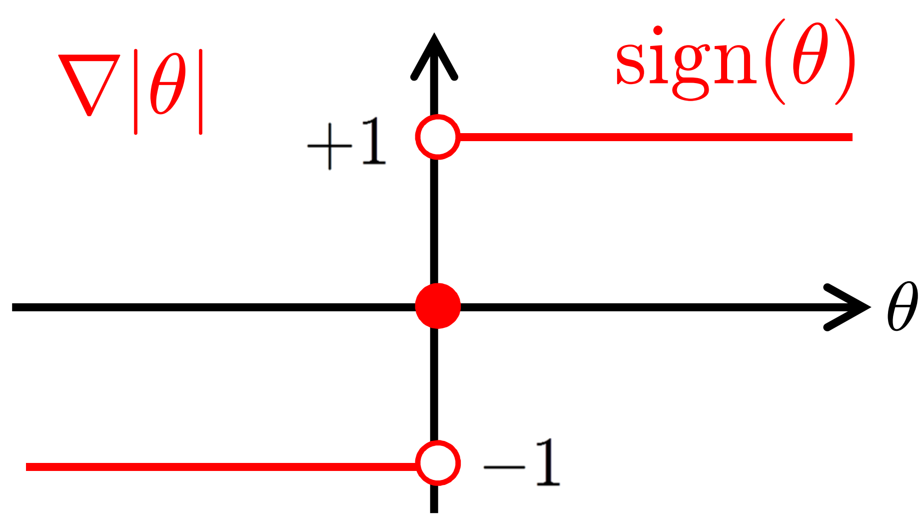
近似的勾配法

- $\boldsymbol{\theta} \longleftarrow \boldsymbol{\theta} - \varepsilon \nabla J(\boldsymbol{\theta})$



- 絶対値の微分を近似する





- しかし，多くの解がゼロになるため，実際には不安定でうまくいかない

交互方向乗数法

- 最適化問題: $\min_{\boldsymbol{\theta}, \boldsymbol{z}} [f(\boldsymbol{\theta}) + g(\boldsymbol{z})]$

  $\text{subject to } \boldsymbol{A\theta} + \boldsymbol{Bz} = \boldsymbol{c}$

- 拡張ラグランジュ関数:

$$L(\boldsymbol{\theta}, \boldsymbol{z}, \boldsymbol{u}) = f(\boldsymbol{\theta}) + g(\boldsymbol{z}) + \boldsymbol{u}^\top (\boldsymbol{A\theta} + \boldsymbol{Bz} - \boldsymbol{c})$$
$$+ \frac{1}{2} \|\boldsymbol{A\theta} + \boldsymbol{Bz} - \boldsymbol{c}\|^2$$

- 解の更新式:
  - $\boldsymbol{\theta}^{(k+1)} = \operatorname*{argmin}_{\boldsymbol{\theta}} L(\boldsymbol{\theta}, \boldsymbol{z}^{(k)}, \boldsymbol{u}^{(k)})$
  - $\boldsymbol{z}^{(k+1)} = \operatorname*{argmin}_{\boldsymbol{z}} L(\boldsymbol{\theta}^{(k+1)}, \boldsymbol{z}, \boldsymbol{u}^{(k)})$
  - $\boldsymbol{u}^{(k+1)} = \boldsymbol{u}^{(k)} + \boldsymbol{A\theta}^{(k+1)} + \boldsymbol{Bz}^{(k+1)} - \boldsymbol{c}$

交互方向乗数法

- 最適化問題: $\min_{\boldsymbol{\theta}, \boldsymbol{z}} \left[f(\boldsymbol{\theta}) + g(\boldsymbol{z}) \right]$

$$\text{subject to } \boldsymbol{A}\boldsymbol{\theta} + \boldsymbol{B}\boldsymbol{z} = \boldsymbol{c}$$

- 拡張ラグランジュ関数:

$$L(\boldsymbol{\theta}, \boldsymbol{z}, \boldsymbol{u}) = f(\boldsymbol{\theta}) + g(\boldsymbol{z}) + \boldsymbol{u}^\top (\boldsymbol{A}\boldsymbol{\theta} + \boldsymbol{B}\boldsymbol{z} - \boldsymbol{c})$$

$$+ \frac{1}{2} \| \boldsymbol{A}\boldsymbol{\theta} + \boldsymbol{B}\boldsymbol{z} - \boldsymbol{c} \|^2$$

θとzが動きすぎると大きくなる

- $\boldsymbol{\theta}^{(k+1)} = \operatorname*{argmin}_{\boldsymbol{\theta}} L(\boldsymbol{\theta}, \boldsymbol{z}^{(k)}, \boldsymbol{u}^{(k)})$
- $\boldsymbol{z}^{(k+1)} = \operatorname*{argmin}_{\boldsymbol{z}} L(\boldsymbol{\theta}^{(k+1)}, \boldsymbol{z}, \boldsymbol{u}^{(k)})$
- $\boldsymbol{u}^{(k+1)} = \boldsymbol{u}^{(k)} + \boldsymbol{A}\boldsymbol{\theta}^{(k+1)} + \boldsymbol{B}\boldsymbol{z}^{(k+1)} - \boldsymbol{c}$

交互方向乗数法

- 最適化問題: $\min_{\boldsymbol{\theta}, \boldsymbol{z}} \left[f(\boldsymbol{\theta}) + g(\boldsymbol{z}) \right]$

$$\text{subject to } \boldsymbol{A\theta} + \boldsymbol{Bz} = \boldsymbol{c}$$

- 拡張ラグランジュ関数:

$$L(\boldsymbol{\theta}, \boldsymbol{z}, \boldsymbol{u}) = f(\boldsymbol{\theta}) + g(\boldsymbol{z}) + \boldsymbol{u}^\top (\boldsymbol{A\theta} + \boldsymbol{Bz} - \boldsymbol{c})$$
$$+ \frac{1}{2} \| \boldsymbol{A\theta} + \boldsymbol{Bz} - \boldsymbol{c} \|^2$$

- 解の更新式:
  - $\boldsymbol{\theta}^{(k+1)} = \underset{\boldsymbol{\theta}}{\mathrm{argm}}$ …
  - $\boldsymbol{z}^{(k+1)} = \underset{\boldsymbol{z}}{\mathrm{argm}}$ …
  - $\boldsymbol{u}^{(k+1)} = \boldsymbol{u}^{(k)} + \boldsymbol{A\theta}^{(k+1)} + \boldsymbol{Bz}^{(k+1)} - \boldsymbol{c}$

等式制約を外れた方向へ
罰則を増やす

ℓ_1-制約付き回帰への適用

- 線形モデル $f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sum_{j=1}^{b} \theta_j \phi_j(\boldsymbol{x})$ における

ℓ_1-正則化最小二乗法の目的関数:

$$\min_{\boldsymbol{\theta}} \left[\frac{1}{2} \|\boldsymbol{\Phi}\boldsymbol{\theta} - \boldsymbol{y}\|^2 + \lambda \|\boldsymbol{\theta}\|_1 \right]$$

$$\boldsymbol{\Phi} = \begin{pmatrix} \phi_1(\boldsymbol{x}_1) & \cdots & \phi_b(\boldsymbol{x}_1) \\ \vdots & \ddots & \vdots \\ \phi_1(\boldsymbol{x}_n) & \cdots & \phi_b(\boldsymbol{x}_n) \end{pmatrix}$$

ℓ_1-制約付き回帰への適用

- 線形モデル $f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sum_{j=1}^{b} \theta_j \phi_j(\boldsymbol{x})$ における

ℓ_1-正則化最小二乗法の目的関数:

$$\min_{\boldsymbol{\theta}} \left[\frac{1}{2} \|\boldsymbol{\Phi}\boldsymbol{\theta} - \boldsymbol{y}\|^2 + \lambda \|\boldsymbol{\theta}\|_1 \right]$$

- これは以下と等価

$$\min_{\boldsymbol{\theta}, \boldsymbol{z}} \left[\frac{1}{2} \|\boldsymbol{\Phi}\boldsymbol{\theta} - \boldsymbol{y}\|^2 + \lambda \|\boldsymbol{z}\|_1 \right] \quad \text{subject to } \boldsymbol{\theta} = \boldsymbol{z}$$

演習

- 解の更新式:
  - $\boldsymbol{\theta}^{(k+1)} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}}\, L(\boldsymbol{\theta}, \boldsymbol{z}^{(k)}, \boldsymbol{u}^{(k)})$
  - $\boldsymbol{z}^{(k+1)} = \underset{\boldsymbol{z}}{\operatorname{argmin}}\, L(\boldsymbol{\theta}^{(k+1)}, \boldsymbol{z}, \boldsymbol{u}^{(k)})$
  - $\boldsymbol{u}^{(k+1)} = \boldsymbol{u}^{(k)} + \boldsymbol{A}\boldsymbol{\theta}^{(k+1)} + \boldsymbol{B}\boldsymbol{z}^{(k+1)} - \boldsymbol{c}$

$$L(\boldsymbol{\theta}, \boldsymbol{z}, \boldsymbol{u}) = f(\boldsymbol{\theta}) + g(\boldsymbol{z}) + \boldsymbol{u}^\top (\boldsymbol{A}\boldsymbol{\theta} + \boldsymbol{B}\boldsymbol{z} - \boldsymbol{c}) + \|\boldsymbol{A}\boldsymbol{\theta} + \boldsymbol{B}\boldsymbol{z} - \boldsymbol{c}\|^2/2$$

- $f(\boldsymbol{\theta}) = \|\boldsymbol{\Phi}\boldsymbol{\theta} - \boldsymbol{y}\|^2,\ g(\boldsymbol{z}) = \lambda\|\boldsymbol{z}\|_1,\ \boldsymbol{A} = -\boldsymbol{B} = I$
  $\boldsymbol{c} = \boldsymbol{0}$ に対して $\boldsymbol{\theta}$ と $\boldsymbol{u}$ の更新式を導出せよ

解答＋宿題

- $$\frac{\partial L}{\partial \boldsymbol{\theta}} = \boldsymbol{\Phi}^\top(\boldsymbol{\Phi}\boldsymbol{\theta} - \boldsymbol{y}) + \boldsymbol{u} + \boldsymbol{\theta} - \boldsymbol{z} = \boldsymbol{0}$$

$$\Rightarrow \boldsymbol{\theta}^{(k+1)} = \operatorname*{argmin}_{\boldsymbol{\theta}} L(\boldsymbol{\theta}, \boldsymbol{z}^{(k)}, \boldsymbol{u}^{(k)})$$
$$= (\boldsymbol{\Phi}^\top\boldsymbol{\Phi} + \boldsymbol{I})^{-1}(\boldsymbol{\Phi}^\top\boldsymbol{y} + \boldsymbol{z}^{(k)} - \boldsymbol{u}^{(k)})$$

- $\boldsymbol{z}$についての更新式(証明は宿題):

$$\boldsymbol{z}^{(k+1)} = \operatorname*{argmin}_{\boldsymbol{z}} L(\boldsymbol{\theta}^{(k+1)}, \boldsymbol{z}, \boldsymbol{u}^{(k)})$$
$$= \max(\boldsymbol{0}, \boldsymbol{\theta}^{(k+1)} + \boldsymbol{u}^{(k)} - \lambda\boldsymbol{1})$$
$$+ \min(\boldsymbol{0}, \boldsymbol{\theta}^{(k+1)} + \boldsymbol{u}^{(k)} + \lambda\boldsymbol{1})$$

要素毎の最大・最小

- また, $\boldsymbol{u}^{(k+1)} = \boldsymbol{u}^{(k)} + \boldsymbol{\theta}^{(k+1)} - \boldsymbol{z}^{(k+1)}$

線形モデルを用いた ℓ_1-正則化最小二乗法に対する交互方向乗数法

- 適当な初期値から，以下の更新を繰り返す：

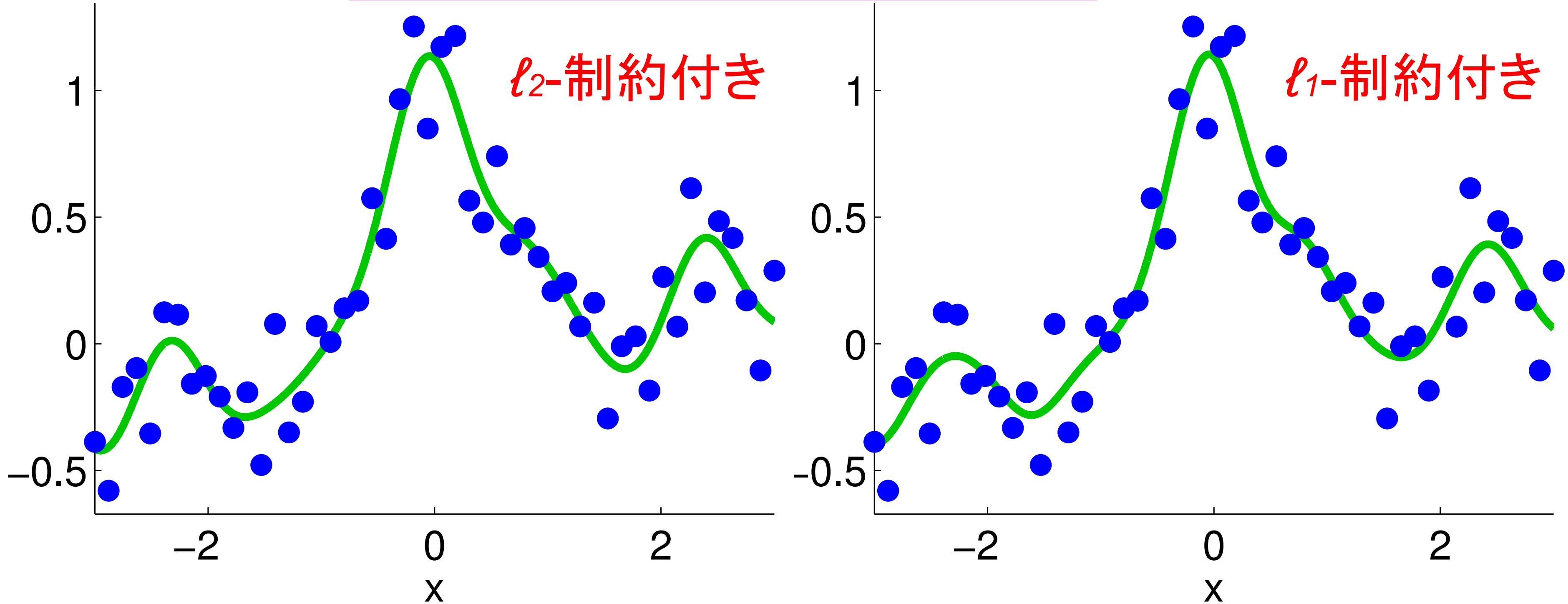
  - $\boldsymbol{\theta} \longleftarrow (\boldsymbol{\Phi}^\top \boldsymbol{\Phi} + \boldsymbol{I})^{-1}(\boldsymbol{\Phi}^\top \boldsymbol{y} + \boldsymbol{z} - \boldsymbol{u})$

  - $$\boldsymbol{z} \longleftarrow \max(\boldsymbol{0}, \boldsymbol{\theta} + \boldsymbol{u} - \lambda\boldsymbol{1}) - \max(\boldsymbol{0}, -\boldsymbol{\theta} - \boldsymbol{u} - \lambda\boldsymbol{1})$$

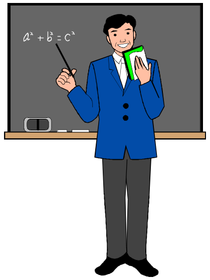
  - $\boldsymbol{u} \longleftarrow \boldsymbol{u} + \boldsymbol{\theta} - \boldsymbol{z}$

実行例

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sum_{j=1}^{n} \theta_j \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{x}_j\|^2}{2h^2}\right)$$



- ℓ_1-制約の結果はℓ_2-制約と見た目は大体同じ
- しかし，ℓ_1-制約は50個のパラメータのうち38個が0



講義の流れ

1. ℓ_1-制約付き最小二乗回帰
2. 解の計算法
3. 様々な拡張
   - A) オンライン学習
   - B) 一般化ℓ_1-ノルム
   - C) ℓ_p-ノルム
   - D) $\ell_1+\ell_2$-ノルム
   - E) $\ell_{1,2}$-ノルム
   - F) トレースノルム

確率的勾配法によるオンライン学習（最小二乗法）

$$\min_{\boldsymbol{\theta}} \frac{1}{2}\sum_{i=1}^{n}\Big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)-y_i\Big)^2$$

$$f_{\boldsymbol{\theta}}(\boldsymbol{x})=\sum_{j=1}^{b}\theta_j\phi_j(\boldsymbol{x})$$

1. パラメータ $\boldsymbol{\theta}$ を適当に初期化
2. 標本 $(\boldsymbol{x}, y)$ をランダムに選び，勾配を少し降下

$$\boldsymbol{\theta} \longleftarrow \boldsymbol{\theta} - \varepsilon \frac{\partial}{\partial \boldsymbol{\theta}} \frac{(f_{\boldsymbol{\theta}}(\boldsymbol{x})-y)^2}{2}$$

$$= \boldsymbol{\theta} - \varepsilon \boldsymbol{\phi}(\boldsymbol{x})\Big(\boldsymbol{\theta}^\top \boldsymbol{\phi}(\boldsymbol{x}) - y\Big)$$

ステップ幅 $\varepsilon > 0$

3. 収束するまで 2. を繰り返す

■どうやってℓ_1-制約条件を満たすか？

スパースオンライン学習

$$\min_{\boldsymbol{\theta}} \frac{1}{2}\sum_{i=1}^{n}\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) - y_i\right)^2 + \lambda\sum_{j=1}^{b}|\theta_j|$$

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sum_{j=1}^{b}\theta_j\phi_j(\boldsymbol{x})$$

1. パラメータ $\boldsymbol{\theta}$ を適当に初期化
2. 標本 $(\boldsymbol{x}, y)$ をランダムに選び，勾配を少し降下

$$\boldsymbol{\theta} \longleftarrow \boldsymbol{\theta} - \varepsilon\boldsymbol{\phi}(\boldsymbol{x})\left(\boldsymbol{\theta}^\top\boldsymbol{\phi}(\boldsymbol{x}) - y\right)$$

ステップ幅 $\varepsilon > 0$

3. 解をスパースにする:

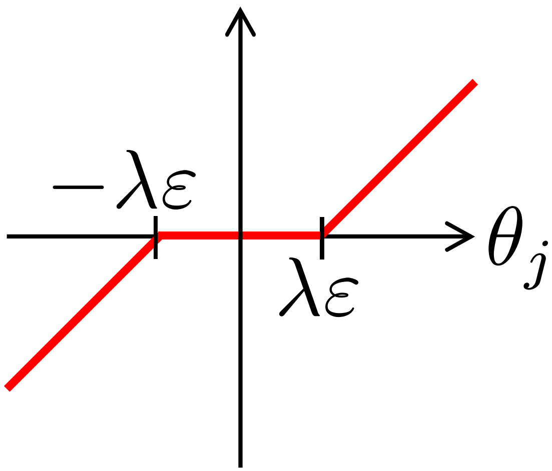
$$\forall j = 1, \ldots, b, \quad \theta_j \longleftarrow \begin{cases} \max(0, \theta_j - \lambda\varepsilon) & (\theta_j > 0), \\ \min(0, \theta_j + \lambda\varepsilon) & (\theta_j \le 0). \end{cases}$$

4. 収束するまで 2. を繰り返す

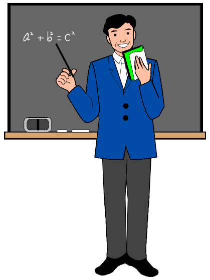
スパース化のイメージ

$$\theta_j \longleftarrow \begin{cases} \max(0, \theta_j - \lambda\varepsilon) & (\theta_j > 0), \\ \min(0, \theta_j + \lambda\varepsilon) & (\theta_j \leq 0). \end{cases}$$

$$\max(0, \theta_j - \lambda\varepsilon) + \min(0, \theta_j + \lambda\varepsilon)$$



- 正則化により原点に引き寄せられる



講義の流れ

1. ℓ_1-制約付き最小二乗回帰
2. 解の計算法
3. 様々な拡張
   - A) オンライン学習
   - B) 一般化ℓ_1-ノルム
   - C) ℓ_p-ノルム
   - D) ℓ_1+ℓ_2-ノルム
   - E) $\ell_{1,2}$-ノルム
   - F) トレースノルム

一般化ℓ_1-ノルム

$$\|\boldsymbol{F}\boldsymbol{\theta}\|_1 = \sum_j \left| \sum_{j'} F_{j,j'} \theta_{j'} \right|$$

■ 例:差分

$$\sum_j |\theta_{j+1} - \theta_j|$$

$$F_{j,j'} = \begin{cases} 1 & (j' = j+1) \\ -1 & (j' = j) \\ 0 & (\text{otherwise}) \end{cases}$$

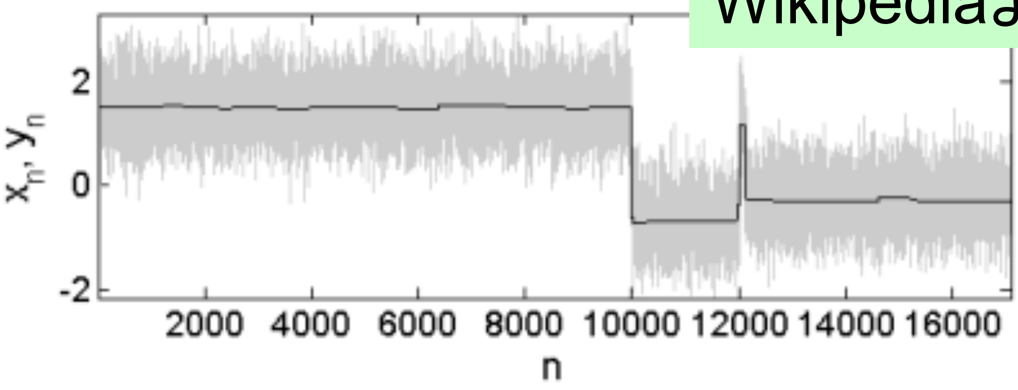
一般化ℓ_1-制約付き最小二乗回帰

$$\min_{\boldsymbol{\theta}} \frac{1}{2}\sum_{i=1}^{n}\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) - y_i\right)^2 \quad \text{subject to} \quad \|\boldsymbol{F}\boldsymbol{\theta}\|_1 \leq R$$

$$R \geq 0$$

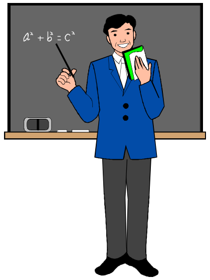
- **例**：全変動ノイズ除去

$$\min_{\boldsymbol{\theta}} \frac{1}{2}\|\boldsymbol{\theta} - \boldsymbol{y}\|^2 \quad \text{subject to} \quad \sum_{j}|\theta_{j+1} - \theta_j| \leq R$$



Wikipediaより

$$F_{j,j'} = \begin{cases} 1 & (j' = j+1) \\ -1 & (j' = j) \\ 0 & (\text{otherwise}) \end{cases}$$

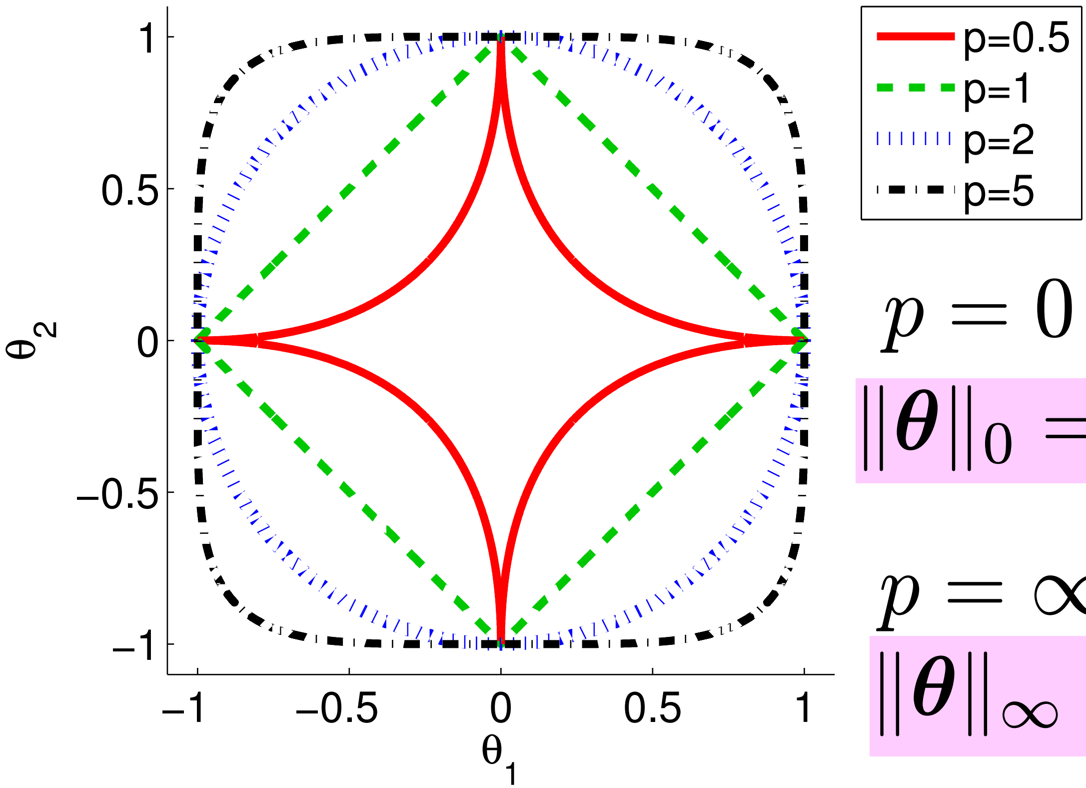


講義の流れ

1. ℓ_1-制約付き最小二乗回帰
2. 解の計算法
3. 様々な拡張
   A) オンライン学習
   B) 一般化ℓ_1-ノルム
   C) ℓ_p-ノルム
   D) ℓ_1+ℓ_2-ノルム
   E) $\ell_{1,2}$-ノルム
   F) トレースノルム

ℓ_p-ノルム

$$\|\boldsymbol{\theta}\|_p = \left(\sum_{j=1}^{b} |\theta_j|^p \right)^{\frac{1}{p}}$$



$$p = 0$$

$$\|\boldsymbol{\theta}\|_0 = \#\ \text{non-zero elements}$$

$$p = \infty$$

$$\|\boldsymbol{\theta}\|_\infty = \max\left\{|\theta_1|, \ldots, |\theta_b|\right\}$$

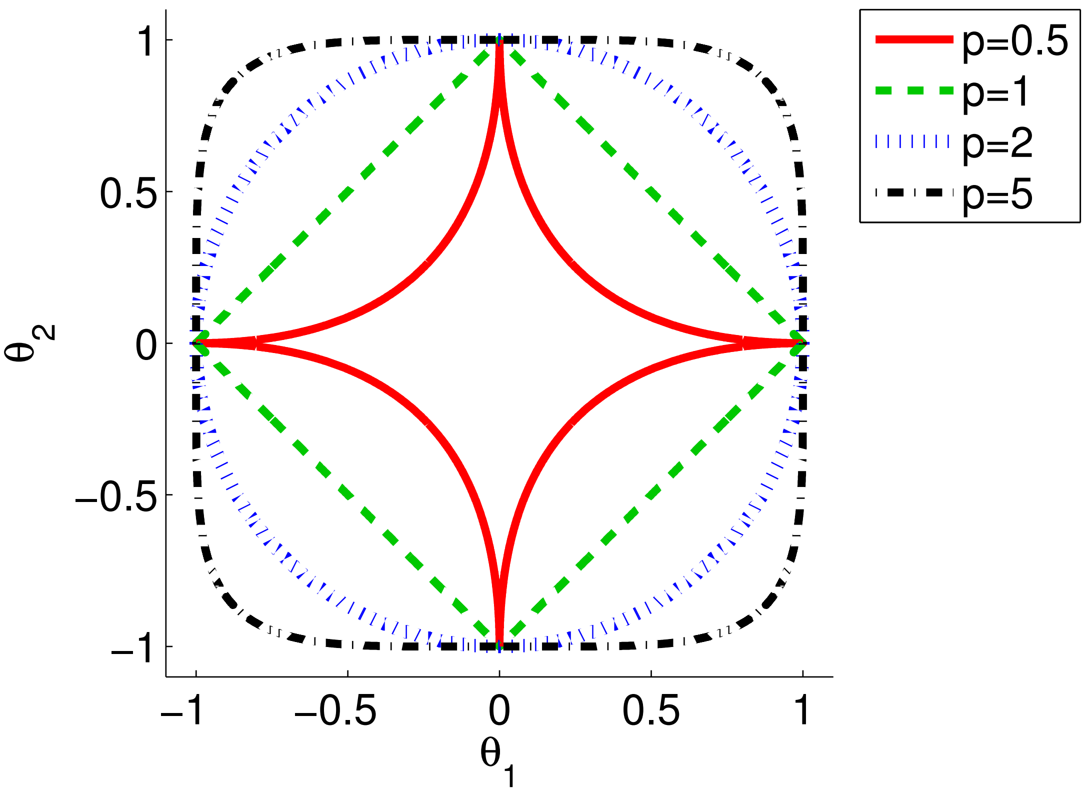
ℓ_p-制約付き最小二乗回帰

- モデルをℓ_p-超球に限定する

$$p \geq 0$$

$$\min_{\boldsymbol{\theta}} \frac{1}{2}\sum_{i=1}^{n}\Big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) - y_i\Big)^2 \quad \text{subject to } \|\boldsymbol{\theta}\|_p \leq R$$

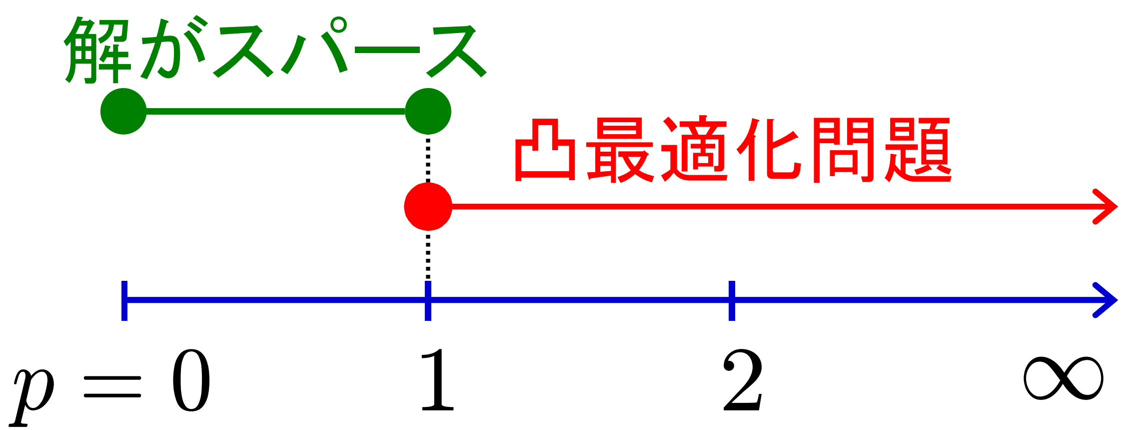
$$R \geq 0$$

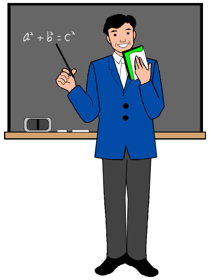


$$\|\boldsymbol{\theta}\|_p = \left(\sum_{j=1}^{b}|\theta_j|^p\right)^{\frac{1}{p}}$$

ℓ_p-制約付き最小二乗回帰

- 解がスパースになる: $0 \leq p \leq 1$
- 最適化問題が凸: $p \geq 1$
（大域的最適解が容易に求まる）
- 両方を満たすのは $p = 1$ のみ！





講義の流れ

1. ℓ_1-制約付き最小二乗回帰
2. 解の計算法
3. 様々な拡張
   - A) オンライン学習
   - B) 一般化ℓ_1-ノルム
   - C) ℓ_p-ノルム
   - D) ℓ_1+ℓ_2-ノルム
   - E) $\ell_{1,2}$-ノルム
   - F) トレースノルム

ℓ_1-制約付き最小二乗回帰の問題点

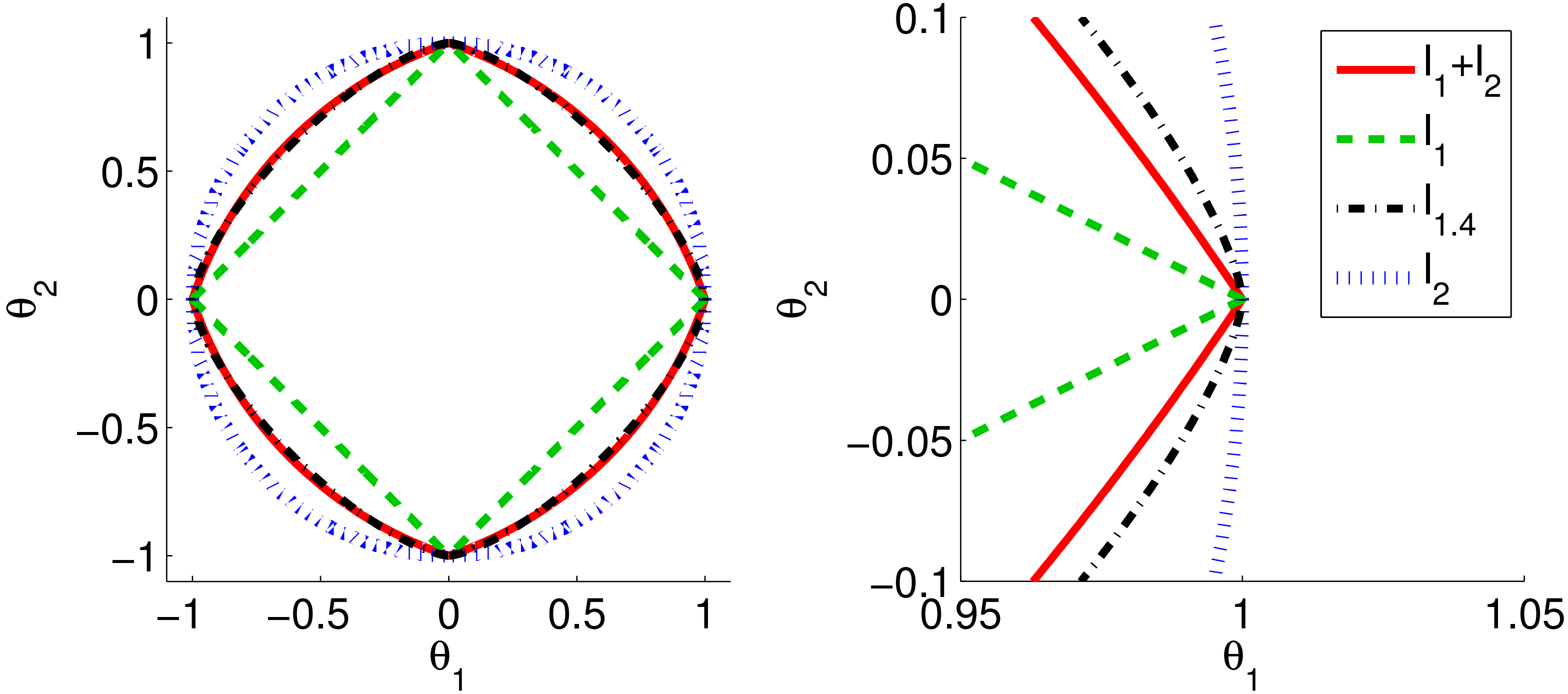
$$\min_{\boldsymbol{\theta}} \frac{1}{2}\sum_{i=1}^{n}\left(\sum_{j=1}^{b}\theta_j\phi_j(\boldsymbol{x}_i)-y_i\right)^2 \quad \text{subject to} \sum_{j=1}^{b}|\theta_j| \leq R$$

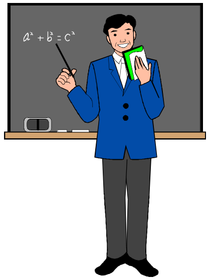
- 非ゼロのパラメータ数はn以下（$b > n$だと不便）
  - 理由：誤差ベクトル $\boldsymbol{\Phi\theta} - \boldsymbol{y}$ における$\boldsymbol{\Phi}$が横長行列
- いくつかの基底関数 $\{\phi_j(\boldsymbol{x})\}_{j=1}^{b}$ が似ている時，それらの一つしか選ばれない
- $b < n$の時，ℓ_2-制約付き最小二乗回帰より性能がやや劣ると言われている

$\ell_1+\ell_2$-制約付き最小二乗回帰

$$(1-\tau)\sum_{j=1}^{b}|\theta_j| + \tau\sum_{j=1}^{b}\theta_j^2 \le R \qquad 0 \le \tau < 1$$

- $\ell_{1.4}$-球と似ているが，$\ell_1+\ell_2$-球は尖っている





講義の流れ

1. ℓ_1-制約付き最小二乗回帰
2. 解の計算法
3. 様々な拡張
   A) オンライン学習
   B) 一般化ℓ_1-ノルム
   C) ℓ_p-ノルム
   D) $\ell_1+\ell_2$-ノルム
   E) $\ell_{1,2}$-ノルム
   F) トレースノルム

$\ell_{1,2}$-ノルム

- 設定：似た性質の変数グループがあるが，それらが役に立つかはわからない
- パラメータ $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_b)^\top$ が

$$\boldsymbol{\theta} = (\boldsymbol{\theta}^{(1)\top}, \ldots, \boldsymbol{\theta}^{(t)\top})^\top \qquad \boldsymbol{\theta}^{(j)} \in \mathbb{R}^{b_j}$$

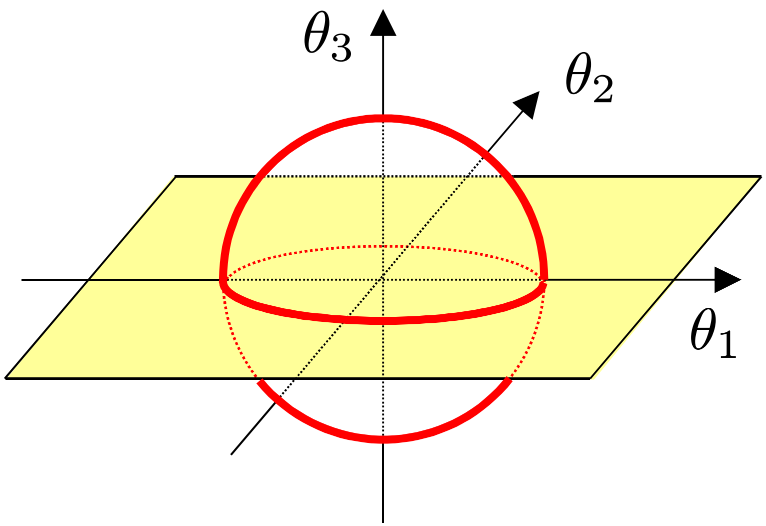
というグループ構造を持っているとき，

$$\|\boldsymbol{\theta}\|_{1,2} = \sum_{j=1}^{t} \|\boldsymbol{\theta}^{(j)}\|_2$$

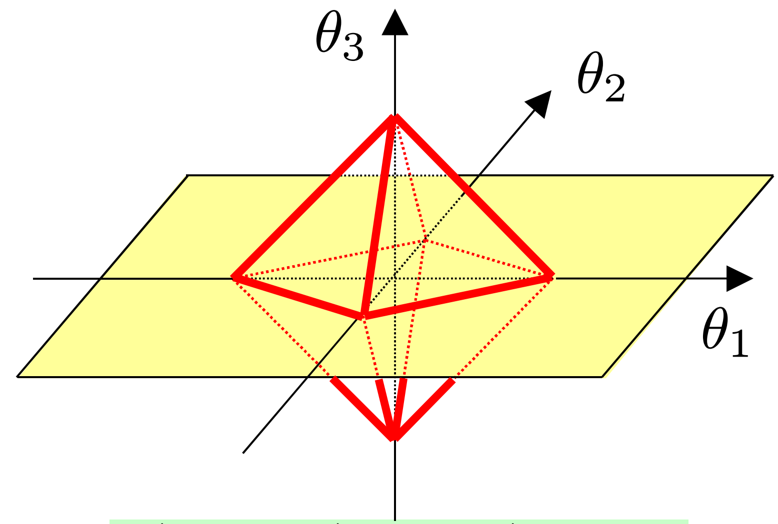
を$\ell_{1,2}$-ノルムとよび，このようなノルムで正則化することをグループ正則化という

演習

- 3次元のℓ_2-ノルムとℓ_1-ノルムの制約条件は，以下のように図示できる．



$$\sqrt{\theta_1^2+\theta_2^2+\theta_3^2}\leq R$$

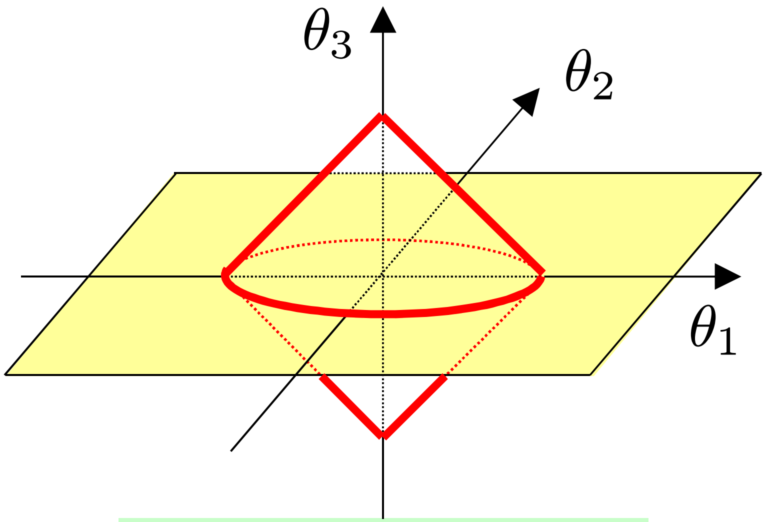


$$\sqrt{\theta_1^2}+\sqrt{\theta_2^2}+\sqrt{\theta_3^2}\leq R$$

- 同様に以下の$\ell_{1,2}$-ノルムを図示し，どのようなスパース解が得られるか説明せよ

$$\sqrt{\theta_1^2+\theta_2^2}+\sqrt{\theta_3^2}\leq R$$

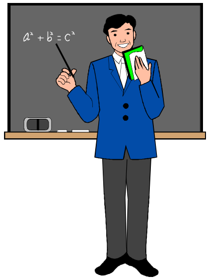
解答例



$$\sqrt{\theta_1^2+\theta_2^2}+\sqrt{\theta_3^2} \le R$$

- グループごとにスパースな解が得られる.

$$\sum_{j=1}^{t} \|\boldsymbol{\theta}^{(j)}\| \le R$$



講義の流れ

1. ℓ_1-制約付き最小二乗回帰
2. 解の計算法
3. 様々な拡張
   - A) オンライン学習
   - B) 一般化ℓ_1-ノルム
   - C) ℓ_p-ノルム
   - D) $\ell_1+\ell_2$-ノルム
   - E) $\ell_{1,2}$-ノルム
   - F) トレースノルム

行列データの回帰

- 行列データ: $\{(\boldsymbol{X}_i, y_i) \mid \boldsymbol{X}_i \in \mathbb{R}^{d_1 \times d_2}, y_i \in \mathbb{R}\}_{i=1}^{n}$
  - 例えば画像や時系列データが入力
- 行列回帰モデル:

$$f_{\boldsymbol{\Theta}}(\boldsymbol{X}) = \mathrm{tr}\left(\boldsymbol{\Theta}^\top \boldsymbol{X}\right) = \sum_{j_1=1}^{d_1} \sum_{j_2=1}^{d_2} \theta_{j_1 j_2} (X)_{j_1 j_2}$$

  - これそのものは，ベクトル化した線形モデルと等価:

$$f_{\boldsymbol{\Theta}}(\boldsymbol{X}) = \mathrm{vec}(\boldsymbol{\Theta})^\top \mathrm{vec}(\boldsymbol{X})$$

  - ℓ_1-ノルムやℓ_2-ノルムで正則化しても，
    行列構造を生かせない

トレースノルム

- 行列 $\mathbf{\Theta} \in \mathbb{R}^{d_1 \times d_2}$ に対するトレースノルム:

$$\|\mathbf{\Theta}\|_{\mathrm{tr}} = \sum_{k=1}^{\min(d_1,d_2)} \sigma_k$$

$\sigma_k \geq 0$: 行列 $\mathbf{\Theta}$ の特異値

- 特異値: $\mathbf{\Theta}^\top \mathbf{\Theta}$ の固有値の平方根
（正方行列なら $\mathbf{\Theta}$ の固有値の絶対値）
- トレースノルム: 特異値に対するℓ_1-ノルム

- 行列のランクは特異値の非ゼロ要素の数に等しい

トレースノルム制約付き最小二乗回帰

$$\min_{\mathbf{\Theta} \in \mathbb{R}^{d_1 \times d_2}} \frac{1}{2} \sum_{i=1}^{n} \left(y_i - \mathrm{tr}\left(\mathbf{\Theta}^\top \boldsymbol{X}_i \right) \right)^2 \ \text{subject to} \ \|\mathbf{\Theta}\|_{\mathrm{tr}} \leq R$$

- トレースノルム制約は,
  特異値に対するℓ_1-ノルム制約
  →特異値がスパースになる
  →行列$\mathbf{\Theta}$ が低ランクになる！

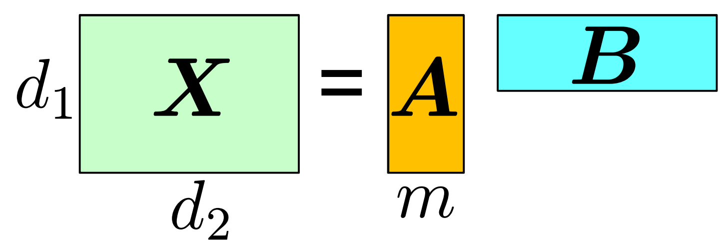
トレースノルムの使用例

- 推薦システムにおいて未知の評価を補完したい

	映画1	映画2	映画3	映画4	$\cdots$
ユーザ1	1	4	*	*	
ユーザ2	*	3	5	2	
ユーザ3	2	*	4	3	
$\vdots$					

- 仮定:各ユーザの評価$(X_{i,1}, X_{i,2}, \ldots, X_{i,d_2})$は$m$人の仮想ユーザの評価$(B_{k,1}, B_{k,2}, \ldots, B_{k,d_2})$の線形結合で近似可能(=低ランク表現可能)

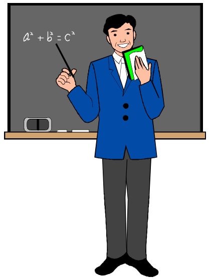
$$X_{i,\cdot} = \sum_{k=1}^{m} A_{i,k} B_{k,\cdot}$$



- トレースノルム正則化により補完が可能

スパース回帰：まとめ

- ℓ_1-正則化学習により解をスパースにする
- 解析的に解は求まらない
- スパース回帰の学習アルゴリズムは盛んに研究されており，最新の手法を用いれば，ℓ_2-正則化学習よりも高速に解ける
  - ゼロを初期値として，非零になる要素を見つけながら学習すれば，最終的に零になる要素をほぼ見ることなく学習できる



次回の予告

- ロバスト回帰（6章）

宿題1

- 次式を証明せよ

$$\operatorname*{argmin}_z T(z) = \max(0, \theta + u - \lambda) + \min(0, \theta + u + \lambda)$$

（この式はソフト閾値処理とよばれる）

$$\lambda \geq 0$$

$$T(z) = \lambda|z| + u(\theta - z) + \frac{1}{2}(\theta - z)^2$$

宿題2

- ガウスカーネルモデル

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sum_{j=1}^{n} \theta_j K(\boldsymbol{x}, \boldsymbol{x}_j)$$

$$K(\boldsymbol{x}, \boldsymbol{c}) = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{c}'\|^2}{2h^2}\right)$$

に対して，スパース回帰の交互方向乗数法による反復式を求め，適当なデータとモデル（例えば前回の三角多項式基底による回帰）に対してスパース回帰を実行せよ